

PC Watchdog Timer Card CE

Model PCWDT

Documentation Number PCWDT2901

This product designed and manufactured in Ottawa, Illinois USA

of domestic and imported parts by

B & B Electronics

International Headquarters

B&B Electronics Mfg. Co. Inc.

707 Dayton Road -- P.O. Box 1040 -- Ottawa, IL 61350 USA

Phone (815) 433-5100 -- General Fax (815) 433-5105

Home Page: www.bb-elec.com

Sales e-mail: orders@bb-elec.com -- Fax (815) 433-5109

Technical Support e-mail: support@bb-elec.com -- Fax (815) 433-5104

European Headquarters

B&B Electronics Ltd.

Westlink Commercial Park, Oranmore, Co. Galway, Ireland

Phone +353 91-792444 -- Fax +353 91-792445

Home Page: www.bb-europe.com

Sales e-mail: orders@bb-europe.com

Technical Support e-mail: support@bb-europe.com

© B&B Electronics -- Revised September 2001

Table of Contents

CHAPTER 1: GENERAL INFORMATION.....	1
INTRODUCTION	1
FEATURES.....	1
SPECIFICATIONS	1
CHAPTER 2: SETUP AND INSTALLATION.....	2
INSPECTION.....	2
ADDRESS SWITCH SETUP	2
JUMPER SETUP.....	5
INSTALLATION	6
CONNECTING THE MOTHERBOARD RESET OPTION	6
INSTALLING THE SOFTWARE	7
CHAPTER 3: OPERATION.....	9
COMMUNICATING WITH THE PCWDT.....	9
PINOUTS.....	10
CHAPTER 4: SOFTWARE.....	11
WINDOWS NT AND WINDOWS 95	11
<i>Changing the I/O Port Address.....</i>	<i>12</i>
<i>Disk Caching.....</i>	<i>12</i>
<i>Existing DOS Programs.....</i>	<i>12</i>
WINDOWS 3.11	12
<i>Changing the I/O Port Address.....</i>	<i>13</i>
<i>Disk Caching.....</i>	<i>13</i>
COMMAND REFERENCE	14
<i>BBPCWDT_Startup</i>	<i>14</i>
<i>BBPCWDT_Shutdown</i>	<i>14</i>
<i>BBPCWDT_Reset</i>	<i>15</i>
APPENDIX A: HARDWARE I/O MAP.....	A-1
APPENDIX B: MODIFYING THE TIMEOUT PERIOD.....	B-1
APPENDIX C: QUICKBASIC EXAMPLE	C-1
APPENDIX D: DECLARATION OF CONFORMITY	D-1

Chapter 1: GENERAL INFORMATION

Introduction

B&B Electronics' PC Watchdog Timer Card, Model PCWDT, is a hardware device designed to overcome the dangers or annoyances associated with a PC "locking up." The PCWDT is periodically reset by your software (or an external trigger). If the watchdog doesn't receive the reset trigger within a software selectable timeout period, the PCWDT will "push" the reset button of the computer. A form C relay output is also provided to reset an external device if desired.

Features

- Software selectable timeout period from 0.5 to 550 seconds. Range can be increased with a simple hardware modification.
- Software enable and disable.
- Form C relay output to control external device.
- Non-splicing connection to motherboard reset pins.
- Isolated timer reset input, for use as alternate trigger source.

Specifications

Dimensions:	2.6" H x 4.1" L
Address:	0 to 3F8h
Power Consumption:	250 mW maximum
Input specs:	Optically isolated input, requires 5 - 12 volts. 50 μ s minimum pulse width
Output specs:	Form C relay output, maximum ratings 24 VDC at 1.0 A 115 VAC at 0.5 A
Connector:	DB-25 male

Chapter 2: SETUP AND INSTALLATION

Inspection

Your PCWDT has been tested for proper operation. It should be in perfect electrical and mechanical condition upon receipt.

Remove the card from its protective packaging. Handle the card only by its edges being careful not to touch the gold connection fingers. Save the packaging for storage or shipping if the card needs repair.

Address Switch Setup

The PCWDT can be set for any I/O address using an eight position DIP switch to program the binary I/O address of the card. Figure 1 is a drawing of the printed circuit board which shows the locations of the address switch and setup jumpers on the PCWDT.

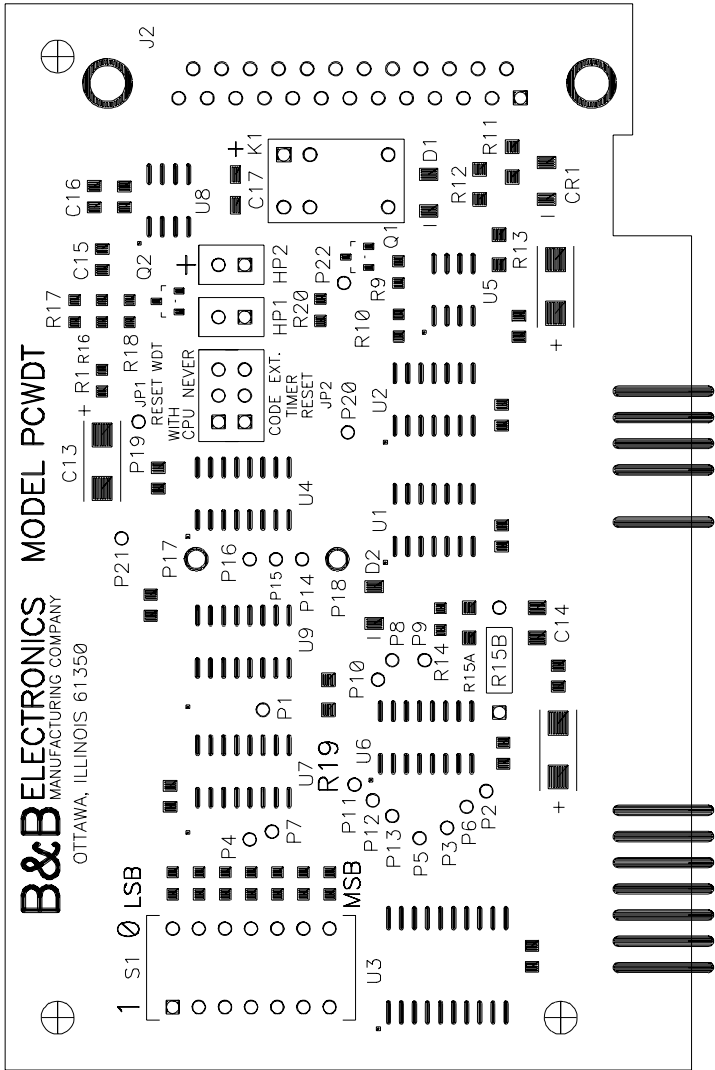


Figure 1. PCWDT PCB Silk Screen

Switch positions 1 - 7 of S1 configure the I/O address of the PCWDT. Switches represent a binary 0 in the ON position, a binary 1 when OFF. Least significant bit (LSB) and most significant bit (MSB) are labeled on the card. Table 1 shows the numerical weight and electrical connection of each switch.

Table 1. Address Switches

S1 Position	7	6	5	4	3	2	1
Bus Connection	SA9	SA8	SA7	SA6	SA5	SA4	SA3
Decimal Weight	512	256	128	64	32	16	8
Hex Weight	200	100	80	40	20	10	8

Any I/O address may be used, but it is important that no other device in the host computer use that address. Table 2 is a list of frequently unused I/O addresses. In most cases, the PCWDT can be set to one of these addresses.

Table 2. Frequently Unused Port Addresses

Base Hex Address	Binary Equivalent	Switch Settings		I/O Space Description
		MSB	LSB	
200	1000000000	1	0	game port
208	1000001000	1	0	game port
300	1100000000	1	1	prototype
308	1100001000	1	1	prototype
310	1100010000	1	0	prototype
318	1100011000	1	0	prototype
380	1110000000	1	1	SDLC
388	1110001000	1	1	SDLC
3A0	1110100000	1	1	bisync comm
3A8	1110101000	1	1	bisync comm

If you want to install at another address, use the following procedure.

1. Select the address: Using an I/O port usage table (one is included in Appendix A) select an unused hex address space. Note that the PCWDT card occupies 8 bytes of I/O space. Use caution when selecting a port address, it is important that nothing else is installed at the selected address.
2. Convert the hex address to its binary equivalent.
3. Throw away the 3 least significant bits.
4. The remaining 7 digits represent the switch address. 1's represent an OFF switch. 0's represent an ON switch.

Jumper Setup

There are three jumpers on the PCWDT that allow the card to be used for many different applications. Refer to Figure 1 or your PCWDT card to locate the three setup jumpers.

JP1 - RESET WDT

JP1 provides two methods of operation after a PCWDT initiated reboot. When JP1 is in the "With Computer" position, the PCWDT will be reset when the host computer is rebooted. Note that when the PCWDT is reset, the relay will return to its normal position. The second mode of operation, "Never" allows the relay to remain energized after the PCWDT reboots the host computer, until the card is addressed by the host software. This mode is useful if using both the reboot and external (relay) reset option on the PCWDT.

JP2 - TIMER RESET

JP2 controls the source of the TIMER RESET signal used to periodically update the PCWDT's counters. This trigger signal can come from the user's program (for example, in the form of a BASIC language OUT statement) or from an external electrical signal. Placing the shorting jumper on the left two pins of JP2 selects the "CODE" or program reset trigger. Placing the shorting jumper on the right two pins selects the "EXTERNAL" trigger option and requires that an external electrical trigger be connected through the DB25 connector. The external trigger must be from 5 to 12 volts and at least 50 μ s in duration.

Installation

1. Turn the power to your computer off.
2. Remove the cover of the computer.
3. Pick an unused slot. The PCWDT will work in a short (8-bit) or full length (16-bit) slot.
4. Remove the expansion slot cover. Save the screw for installation of PCWDT.
5. Install the PCWDT into the unused slot. Be certain that the card is inserted completely into the slot.
6. Secure the card with the mounting screw.

Connecting the Motherboard Reset Option

To use the PCWDT to reboot the host computer a connection must be made from the PCWDT to the motherboard. The jumper wire supplied with the PCWDT must be connected between the HP1 pins on the PCWDT and the RESET pins on the host motherboard. If your computer has a RESET switch, the switch's jumper wires should already be connected to the motherboard reset pins. Remove the switch wires from the motherboard pins and make the connection from the PCWDT to the motherboard RESET pins with the provided wire. Then connect the switch wires to HP2 of the PCWDT. These pins are simply in parallel with HP1 and will allow the PCWDT or your RESET switch to reboot the host computer.

The polarity of HP1 and HP2 are indicated on the PCWDT PCB but, in most cases, not on the motherboard or reset switch. If you are unsure of which way to connect the PCWDT to the motherboard, use the following procedure to verify the polarity of the connections.

1. Disconnect the reset switch wires from the motherboard.
2. Install the PCWDT in an empty slot in your computer. Leave the computer cover off.
3. Connect the jumper wire provided with the PCWDT to the motherboard reset pins. Leave the other end unconnected.
4. Power up the computer.
5. After the computer has booted, connect the other end of the jumper wire to HP1 of the PCWDT.
6. Remove the wires from HP1.

7. If your computer does not reboot, the polarity was correct. If the computer did reboot, reverse the connection made to HP1 of the PCWDT.
8. Follow the same procedure to connect the RESET switch wires to HP2 of the PCWDT.

NOTE: Use caution when operating the computer with the case open. Be sure to ground yourself by touching the metal case of the computer before you handle any components inside the machine.

Installing the Software

The software provided with the PCWDT is only for Windows. If you are not using Windows, you do not need to install the software.

1. Insert the floppy disk in drive A:
2. Run the setup program. This step is different depending on the version of Windows:
 - A. For Windows 3.11, select RUN from the FILE MENU of the Program Manager and type: `A:\SETUP.EXE`
 - B. For Window 95 or Windows NT, select SETTINGS | CONTROL PANEL | ADD/REMOVE PROGRAMS from the Start Button and type `A:\SETUP.EXE`
3. Follow the directions in the setup program to complete the installation of the software.

After the software is installed, the file, FILES.LST, contains a list and description of the installed files. The file, HISTORY.LST, contains a historic description of the product. The file, READ.ME, contains corrections and additions to the printed user's manual.

Chapter 3: OPERATION

Communicating with the PCWDT

The PCWDT uses a four-bit latch to start and stop the timers and set the timeout period.

Regardless of the mode of operation of the PCWDT, a "1" must be written to the least significant bit of the latch to start the timers. Writing a "0" to the least significant bit of the latch at any time will stop the timers. Note that since the least significant bit is used to start and stop the counters, any even number written to the latch will result in the counters being turned off.

The remaining three bits of the latch are used to select one of eight timeout periods. Table 3 shows the possible timeouts.

Table 3. PCWDT Timeouts

Binary Latch Value	Hex Latch Value	Approximate Timeout (seconds)
0001	1	0.5
0011	3	2.1
0101	5	8.6
0111	7	17.2
1001	9	34.4
1011	B	137.9
1101	D	275.8
1111	F	551.6

A simple hardware modification can be made to further extend the timeout range. See Appendix B for information on making this modification.

To use the PCWDT from your program, the latch value corresponding to the desired timeout must be periodically written to the latch. After the counters are started, they are reset every time the PCWDT is addressed. As long as the PCWDT is addressed before the end of the selected timeout period, no reset will occur. To do this from a BASIC program, place an OUT <PCWDT address>, <latch value> somewhere in a reoccurring loop. Each time the OUT statement is executed, the PCWDT counters are reset and begin counting again. To disable the PCWDT, place an OUT <PCWDT address>, <0> in your program. Be certain that this statement is

executed whenever your program is terminated. In Pascal, the *port* command may be used in place of the BASIC OUT statement. In C, the *outp* command is used. Consult your programming languages manual for the syntax used for each command. A small example program, written in QuickBASIC can be found in Appendix C.

To use the external trigger to reset the PCWDT, your software must first start the counters and set the timeout period. This can be done with a single OUT <PCWDT address>, <latch value> statement in your program. Thereafter, a reset pulse must be received through the DB25 connector to reset the counters.

Pinouts

Four pins of the male DB25 connector are used for the PCWDT. Pins 9 and 10 are the external device reset relay contacts. These are non-polarized, normally open contacts. Pins 15 and 16 are the external reset input pins. Figure 2 is a schematic of the I/O portion of the PCWDT.

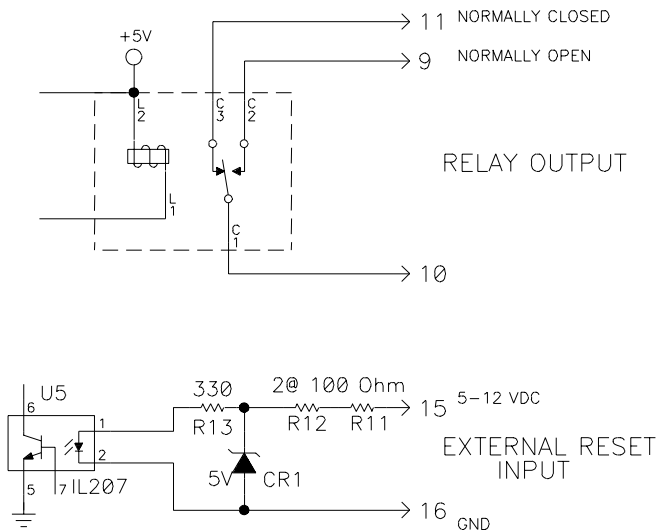


Figure 2. I/O Schematic

Chapter 4: SOFTWARE

The software shipped with the PCWDT includes example software for Windows 3.11, Windows 95 and Windows NT written for Borland C++ 5.01, Microsoft Visual C++ 4.0 and Microsoft Visual BASIC 4.0. Also included is a Windows NT 4.0 device driver.

Windows NT and Windows 95

Windows NT restricts access to the I/O ports of the computer to device drivers, so a simple `OUTP` command cannot be used to communicate with the watchdog timer. A device driver, for use with Windows NT 4.0 or later, is included with the watchdog timer software to allow access to the watchdog timer under Windows NT. A dynamic-link library (DLL) is included to hide the details of communicating with this device driver. These commands are explained in the Command Reference section of this chapter.

Windows 95 is less restrictive than Windows NT on the access of the computer's I/O ports. A simple `OUTP` command can be used in Borland C++ or Microsoft Visual C++ to communicate with the watchdog timer; however, using the DLL will allow the same program to run under both Windows 95 and Windows NT. Visual BASIC does not have a command to access the I/O ports of the computer, so the DLL must be used.

A Visual BASIC program must include definitions for the routines in the DLL in order to call them. The file, `PCWDT.BAS`, contains definitions for these function and can be included in the Visual BASIC project by selecting `ADD FILE` under the `FILE` menu.

In a Borland C++ or Microsoft Visual C++ program, each module calling the watchdog timer routines must include the file, `PCWDT.H`. This file contains the definitions for the functions in the DLL. The file, `BBPCWDT.LIB`, must be included in the project to inform the linker that it is using a DLL. Note that the `BBPCWDT.LIB` file for Borland C++ and Microsoft Visual C++ are not the same, so they are not interchangeable. A console mode program, written in C, is provided to demonstrate the use of the DLL functions. Both Borland C++ and Microsoft Visual C++ project files are include to make it easier to recompile the example program. The file, `MKIMPLIB.BAT`, is provided as an example of how to create the import library, `BBPCWDT.LIB`. It may be necessary to recreate the import library if the application is not compiled with Borland C++ version 5.01 or Microsoft Visual C++ version 4.0.

Changing the I/O Port Address

The setup program, WDTSETUP.EXE, changes the I/O port address that the software uses to communicate with watchdog timer. This address must match the setting of the jumper switches on the watchdog timer. The program updates keys in the registry for the device driver. In order to update these keys under Windows NT Server, the user must be logged in as the administrator or have administrator rights. Windows NT Workstation and Windows 95 allows all users to modify these keys. Under Windows NT, both server and workstation, the changes made by the setup program take effect the next timer the computer is restarted. Under Windows 95, the changes are effective immediately, but any program that is using the watchdog timer when the change is made must be restarted for the changes to take effect.

Disk Caching

Both Windows NT and Windows 95 uses disk caching to speed up access of the hard drive. This means that the operating system decides when to write data to the disk instead of it being done when your application writes it. Rebooting the computer with the watchdog timer with write-behind caching enabled may cause the loss of all data in the write-behind cache.

Existing DOS Programs

The NT device driver does not allow existing DOS programs to communicate with the watchdog timer. DOS programs will work under Windows 95, because Windows 95 allows direct access to the I/O ports of the computer.

Windows 3.11

Windows 3.11, just like DOS, allows full access to the I/O ports of the computer, so a simple OUT command can be used to communicate with the watchdog timer. A 16-bit dynamic-link library (DLL) is supplied to provide a common interface to the watchdog timer as in Windows 95 and Windows NT. The commands are explained in the command reference section of this chapter.

In Borland C++, a program using the watchdog timer must include PCWDT.H and PCWDT16.LIB in the project to use the watchdog timer commands. The example program uses Borland's EasyWin to demonstrate the use of the DLL functions. A Borland C++ project file is include to make it easier to recompile the example program.

Changing the I/O Port Address

The I/O port address that the software uses to communicate with the watchdog timer is stored in the file, PCWDT.INI, located in the Windows directory. Use NOTEPAD to open this file and change the values for the parameter `IoPortAddress`. The default value for `IoPortAddress` is 0x0200 (hexadecimal 200). This value must match the settings of the jumper switches on the watchdog timer. The format of the PCWDT.INI file is:

```
[Parameters]
IoPortAddress=0x0200
IoPortCount=0x0001
```

Currently the value of `IoPortCount` is ignored, but may be used in future versions of this software, so it should not be changed. These changes take effect immediately when they are saved; however, any program that uses the watchdog timer and is running at the time of the change will not recognize the changes until it is restarted. There is no need to reboot the computer for the changes to take effect.

Disk Caching

If you are using SmartDrive, or any other disk caching software, with write behind caching enabled, you may lose all data in the write behind cache by rebooting the computer with the watchdog timer. Disable the write behind cache before running an application that uses the watchdog timer to reboot the computer.

Command Reference

BBPCWDT_Startup

- Purpose:** Begin communications with the watchdog timer.
- C Syntax:** `HANDLE BBPCWDT_Startup(void);`
- BASIC Syntax:** `Function BBPCWDTStartup () As Long`
- Remarks:** This function reads the watchdog timer's I/O port value that the setup program stored in the registry for Windows 95 / NT or in the file, PCWDT.INI, for Windows 3.11. The Windows NT version of this function establishes communications with the PCWDT device driver. The Windows 3.11 / 95 version of this function just return a handle that is used with the other functions in this section to communicate with the watchdog timer.
- Return Value:** On error, `INVALID_HANDLE_VALUE` is returned otherwise a handle to the watchdog timer is returned.

BBPCWDT_Shutdown

- Purpose:** Terminate communications with the watchdog timer.
- C Syntax:** `BOOL BBPCWDT_Shutdown(HANDLE hDev);`
- BASIC Syntax:** `Function BBPCWDTShutdown (ByVal hDev As Long) As Byte`
- Remarks:** *hDev* is the handle to the watchdog timer that was returned by `BBPCWDT_Startup`. The Windows NT version of this function closes the communications to the PCWDT device driver.
- Return Value:** `TRUE` (non-zero) if successful, otherwise `FALSE` (zero).

BBPCWDT_Reset

- Purpose:** Reset the timeout of the watchdog timer.
- C Syntax:** `BOOL BBPCWDT_Reset(HANDLE hDev, BYTE bResetValue);`
- BASIC Syntax:** `Function BBPCWDTReset (ByVal hDev As Long, ByVal bResetValue As Byte) As Byte`
- Remarks:** *hDev* is the handle to the watchdog timer that was returned by `BBPCWDT_Startup`. *bResetValue* is the reset value that is sent to the watchdog timer. The values are:

Value	Timeout
0	disabled
1	0.5 seconds
3	2.1 seconds
5	8.6 seconds
7	17.2 seconds
9	34.4 seconds
11	137.9 seconds
13	275.8 seconds
15	551.6 seconds

Return Value: TRUE (non-zero) if successful, otherwise FALSE (zero).

Appendix A: Hardware I/O Map

I/O Map of XT Class Machines

Hex Address	Address Function in XT Class Machines
000-00F	DMA controller (8237A)
020-021	interrupt controller (8259A)
040-043	timer (8253)
060-063	PPI(8255A)
080-083	DMA page register (74LS612)
0A0-0AF	NMI - non maskable interrupt
200-20F	game port joystick controller
210-217	expansion unit
2E8-2EF	COM4 serial port
2F8-2FF	COM2 serial port
300-31F	prototype card
320-32F	hard disk
378-37F	parallel print
380-38F	SDLC
3B0-3BF	MDA - monochrome adapter and printer
3D0-3D7	CGA - color graphics adapter
3E8-3EF	COM3 serial port
3F0-3F7	floppy diskette controller
3F8-3FF	COM1 serial port

I/O Map of AT Class Machines

Hex Address	Address Function in AT Class Machine
000-01F	DMA controller #1 (8237A-5)
020-03F	interrupt controller #1 (8259A)
040-05F	timer (8254)
060-06F	keyboard (8042)
070-07F	NMI - non maskable interrupt & CMOS RAM
080-09F	DMA page register (74LS612)
0A0-0BF	interrupt controller #2 (8259A)
0C0-0DF	DMA controller #2 (8237A)
0F0-0FF	80287 math coprocessor
1F0-1F8	hard disk
200-20F	game port joystick controller
258-25F	Intel Above Board
278-27F	parallel printer port 2
2E8-2EF	COM4 serial port
2F8-2FF	COM2 serial port
300-31F	prototype card
378-37F	parallel printer 1
380-38F	SDLC or bisynch com 2
3A0-3AF	bisynch com 1
3B0-3BF	MDA - monochrome adapter
3BC-3BE	parallel printer on monochrome adapter
3C0-3CF	EGA - reserved
3D0-3D7	CGA - color graphics adapter
3E8-3EF	COM 3 serial port
3F0-3F7	floppy diskette controller
3F8-3FF	COM1 serial port

Any eight-byte space not listed above or not used by any other equipment in your system may be used by the PCWDT.

Appendix B: Modifying the Timeout Period

The PCWDT printed circuit board (PCB) was designed to make it easy to modify the timeout range if necessary.

The PCWDT's counter circuit can be broken into three basic blocks: the oscillator, which is set to 14.5 kHz, and two divide by X counters. Each of the counters has 10 different outputs. The first counter is connected to divide the oscillator signal by 1024. This divided signal is connected to the input of the second counter where it is again divided and the appropriate timeout output is selected by your software. By selecting a divisor other than 1024 for the first divider, the entire timeout range can be modified.

Locate parts U6 and U7 on the PCWDT. These are the two divide by X counters. R19 is a zero ohm jumper that connects the output of U6 to U7's input. R19 must be removed. By connecting a jumper wire from pad P1 to one of the pads P2-P13 from Table B1 below, a wide range of timeouts can be selected.

Table B1. Divider Outputs

Pin #	Pin Name	Divisor
7	P2	16
5	P3	32
4	P5	64
6	P6	128
14	P8	256
13	P9	512
15	P10	1024
1	P11	4096
2	P12	8192
3	P13	16384

With the connection made, the timeout period corresponding to each of the latch values listed in the table will change by a factor of the new divisor divided by 1024. For example, if the connection is made to Q13 on pin 2, the divisor is now 8192. This would increase the timeouts by a factor of $8192 / 1024 = 8$. What used to be a 3 second timeout would now be 3×8 , or 24 seconds.

Appendix C: QuickBASIC Example

Note: Add additional test program for the PCWDT is available on B&B's ftp site at <ftp.bb-elec.com/bb-elec/software>.

B&B Electronics Mfg. Co. © 1992
PC Watchdog Timer QuickBASIC Example Program

This code is an example of how the PCWDT is addressed from within a program. This code would be used when the TIMER RESET INPUT jumper is in the "CODE" position.

```
PCWDTaddress = &H300           'hex address chosen for the PCWDT

LatchValue = &H7               'hex latch value chosen, in this
                               'case about a 17 second timeout


'Main loop of your program
DO WHILE INKEY$ = ""           'repeat loop until keypressed

OUT PCWDTaddress, LatchValue   'This statement must be executed
                               'within the timeout period (17
                               'seconds for this example) or the
                               'PCWDT will reboot the computer.

LOOP

OUT PCWDTaddress, 0           'disable PCWDT before ending
END
```

APPENDIX D: DECLARATION OF CONFORMITY

DECLARATION OF CONFORMITY	
Manufacturer's Name:	B&B Electronics Manufacturing Company
Manufacturer's Address:	P.O. Box 1040 707 Dayton Road Ottawa, IL 61350 USA
Model Numbers:	PCWDT
Description:	PC Watchdog Timer Card
Type:	Light industrial ITE equipment
Application of Council Directive:	89/336/EEC
Standards:	EN 50082-1:1998 EN 50082-1:1992 IEC 801 (-2, -3, -4) EN 61000 (-4-2, -4-3, -4-4, -4-6)
	
Michael J. Fahrion, Director of Engineering	
